Freescale Semiconductor
Engineering Bulletin

Document Number: EB747
Rev. 1, 06/2011

FlexRay Protocol Command Status

by: Stephan Mueller
Kunal Prasad
David Paterson
Dirk Moeller

1 Introduction

When a protocol command is issued to the Protocol Engine (PE), the application software must ensure that the protocol state change has successfully transitioned before issuing further protocol commands.

This document describes the interpretation and use of the BSY bit and its meaning in the Protocol Operation Control Register (FR_POCR).

The description of the BSY bit, shown below, can give the impression that the BSY bit signals when a new command can be written:

Contents

Table 1. FR_POCR field description

Field	Description
BSY	**Protocol Control Command Write Busy**—This status bit indicates the acceptance of the protocol control command issued by the application via the Protocol Control Command (POCCMD) field in the Protocol Operation Control Register (POCR). The Communication Controller (CC) sets this status bit when the application has issued a protocol control command via the POCCMD field. The CC clears this status bit when a protocol control command was accepted by the Protocol Engine (PE). When the application issues a protocol control command while the BSY bit is asserted, the CC ignores this command, sets the protocol command ignored error flag PCMI_EF in the CHI Error Flag Register (FR_CHIERFR), and will not change the value of the POCCMD field. 0 Command write idle, command accepted and ready to receive new protocol command. 1 Command write busy, command not yet accepted, not ready to receive new protocol command.

This bit should not be used to check if the protocol command has been processed and transitioned to that state.

The physical implementation shows that whenever a command is issued by the host it is immediately (~1 CHI clock) acknowledged by the Controller Host Interface (CHI) and it is not guaranteed that the PE is in a correct state to accept a new command. In this state, it can respond by:

- Asserting the Illegal Protocol Control Command (IPC) error flag in the Protocol Interrupt Flag Register 1 (FR_PIFR1), which could cause an interrupt if enabled via the Protocol Interrupt Enable Register 1 (FR_PIER1).
- Setting the System Bus Communication Failure Error Flag (SBCF_EF) in the CHI Error Flag Register (FR_CHIERFR).
- Eventally transitioning to freeze mode (POC:halt state), which would result in no communication until reconfigured.

Therefore, the application software has to check the protocol state before issuing a new command.

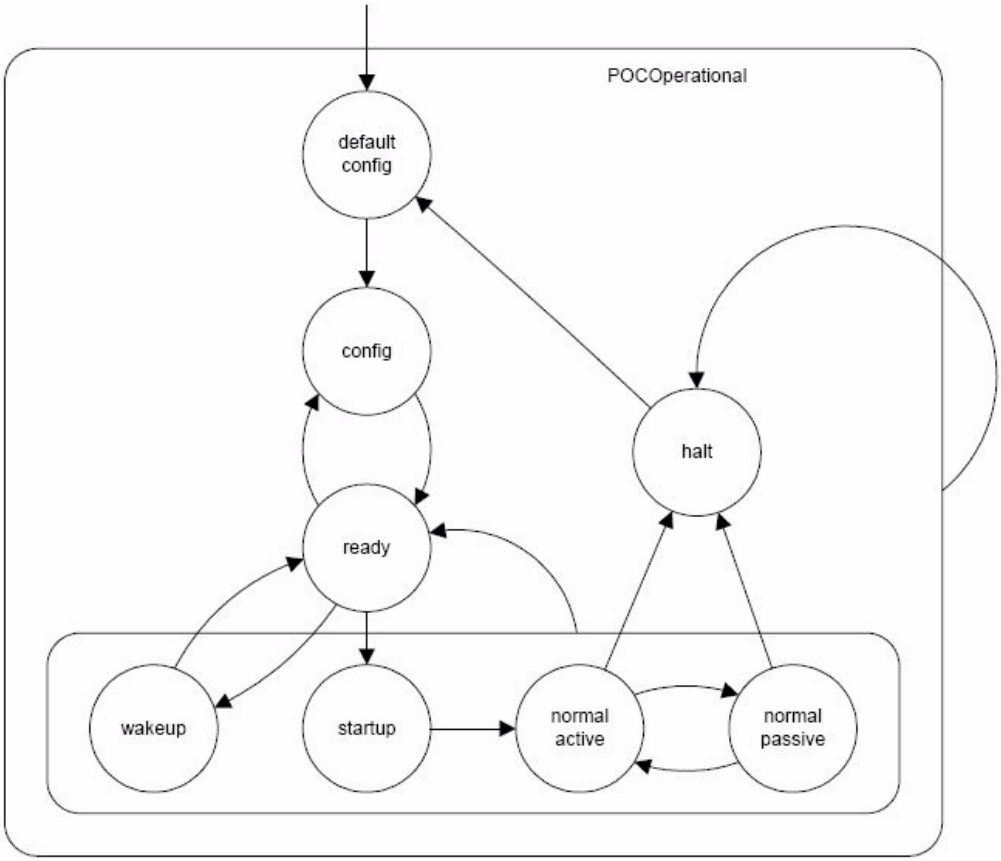
2 FlexRay Protocol Specification

The FlexRay protocol specification (V2.1A) does not give any information on whether the Protocol Control Command Write is busy. It states that the following protocol operation control-related status variable shall be provided in the CHI:

- the status variable vPOC!State (as maintained by the POC process)

It should be interpreted within the Protocol Specification that the application software should monitor the protocol state before transitioning to another state.

There are various state transitions possible:



3 Correct Application Software Implementation

Before issuing the next command, the software needs to check that the previous command or state has been entered properly. This can be checked using the Protocol State Register (PSR).

For example, the following code snippet shows a software test loop to check that READY state has been entered before giving the RUN command:

```
end=FALSE;
endTimeout=FALSE;
/* Wait till Protocol Command Write is not busy */
while ((end==FALSE)  && (endTimeout==FALSE))
{
   if (FR_POCR_BSY != (FR_POCR_Reg) & FR_POCR_BSY))
   {
      /* BSY flag is not set, command was processed */
      end= TRUE;
   }
   if(timeout >= FR_MAX_WAIT_CYCLES)
   {
      /* Timeout occured */
      endTimeout= TRUE;
   }
   timeout++;
}

/* Timeout occured */
if(endTimeout==TRUE)
{
    ...
   /* Error handler */
}
```

```
/*Verify CC is in the POC:Ready state and FRZ bit is not set*/
if ((FR_PSR1_FRZ != (FR_PSR1_Reg) & FR_PSR1_FRZ)  \
     && (FR_PSR0_PROTSTATE_READY == (FR_PSR0_Reg & FR_PSR0_PROTSTATE_MASK))
{
   /* Invoke POCCMD command RUN */
   FR_POCR_reg = (FR_POCR_CMD_RUN |FR_POCR_WME);
}
```

4 Revision history

Table 2. Revision history

Version	Changes
Rev. 0	First public version of this document.
Rev. 1	In Section 3, “Correct Application Software Implementation,” changed “entTimeout=FALSE” to “endTimeout=FALSE.”
Changes made April 2012[1]	Front page: Add SafeAssure branding. Back page: Apply new back page format.

[1] No substantive changes were made to the content of this document; therefore the revision number was not incremented.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number: EB747
Rev. 1
06/2011

freescale™